

## **AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the above-referenced application.

### **Listing of Claims:**

Claims 1 - 65 (Cancelled).

66. (Currently amended) The method of Claim ~~[[63]]~~ 78, wherein said sender is a WAN blade coupled to said first storage device.

67. (Previously Presented) The method of Claim 66, wherein when a failure is determined, journal entries in each of said secondary storage device are determined to be unsynchronized.

68. (Previously Presented) The method of Claim 67, wherein a failure prevents a consistency group of storage devices from synchronizing data, said first storage device and said at least one other secondary storage device being included in said consistency group.

69. (Previously Presented) The method of Claim 68, wherein, upon occurrence of a failure, the consistency group triggers so that the secondary storage devices are deemed not ready by the first storage device and any subsequent updates on the first storage device are indicated as invalid on the secondary storage devices.

70. (Previously Presented) The method of Claim 69, wherein in response to said failure, data marked as invalid on the secondary devices is copied from the first to the other secondary storage devices as part of data restoration.

71. (Currently amended) The method of Claim [[63]] 78, wherein said failure is a link failure occurring when at least one communication link fails.

72. (Previously Presented) The method of Claim 71, further comprising:

detecting a link failure by failure of a linked device to response to a direct inquiry.

73. (Previously Presented) The method of Claim 72, wherein said link failure is a failed link between two WAN blades connected by a network.

74. (Previously Presented) The method of Claim 72, wherein said link failure is a failed link between a WAN blade coupled to a primary storage device.

75. (Previously Presented) The method of Claim 74, further comprising: in response to detecting said failed link, journaling writes to the WAN blade rather than the primary storage device, said WAN blade acting as a buffer to compensate for said failed link.

76. (Previously Presented) The method of Claim 75 , wherein, upon said WAN blade having a journal that overflows, said WAN blade not acknowledging write operations by the primary storage device.

77. (Previously Presented) The method of Claim 76, further comprising:

clearing said journal on said WAN blade in response to said journal overflowing by sending messages to the primary storage device to invalidate previously written and acknowledged data.

78. (Currently amended) ~~[[The]]~~ A method of Claim 63, for performing data recovery in a computer system, comprising:

sending data from a first storage device to at least one other secondary storage device, said data being sent in a plurality of data packets, each of said plurality of packets being associated with a sequence number having a first predetermined value;

upon determining that the data has been successfully stored on all of said at least one other storage device, deleting journal entries in a sender corresponding to said data; and

upon determining a failure in connection with synchronizing data between a first storage device and at least one other secondary storage device, deleting journal entries in each of said at least one other secondary storage device, and resending unsynchronized journal entries from the sender by sending a plurality of data packets all having a same sequence number lower than sequence numbers associated with other unsent packets and then sending any remaining data packets having a next higher sequence number, wherein data packets having the same sequence number are sent in an order that is independent of an order in which the data packets were created and wherein data packets having the same sequence number represent different data that is independently generated from a plurality of different sources,

wherein in response to the sequence number in the sender becoming equal to a second predetermined value different from the first predetermined value, acknowledging receipt of the blocks of data corresponding to the packets of data that are assigned the first predetermined value as the sequence number and sending the packets of data that are assigned the first predetermined value as the sequence number to said at least one other secondary storage device.

79. (Previously Presented) The method of Claim 78, wherein said acknowledging includes sending an acknowledgement to a host in the computer system sending data to the first storage device prior to said data actually being transferred to the at least one secondary storage device.

Claims 80 - 82 (Cancelled).

83. (Currently amended) The computer program product of Claim ~~[[80]]~~ 95, wherein said sender is a WAN blade coupled to said first storage device.

84. (Previously Presented) The computer program product of Claim 83, further comprising machine executable code that, when a failure is determined, determines journal entries in each of said secondary storage device to be unsynchronized.

85. (Previously Presented) The computer program product of Claim 84, further comprising machine executable code that determines failure prevents a consistency group of storage devices from synchronizing data, said first storage device and said at least one other secondary storage device being included in said consistency group.

86. (Previously Presented) The computer program product Claim 85, further comprising machine executable code that, upon occurrence of a failure, triggers a consistency group in that the secondary storage devices are deemed not ready by the first storage device and any subsequent updates on the first storage device are indicated as invalid on the secondary storage devices.

87. (Previously Presented) The computer program product of Claim 86, further comprising machine executable code that, in response to said failure, marks data as invalid on the secondary devices and copies said data from the first storage device to the secondary storage devices as part of data restoration.

88. (Currently amended) The computer program product of Claim ~~[[80]]~~ 95, further comprising machine executable code that determines said failure is a link failure occurring when at least one communication link fails.

89. (Previously Presented) The computer program product of Claim 88, further comprising:  
machine executable code that detects a link failure by failure of a linked device to response to a direct inquiry.

90. (Previously Presented) The computer program of Claim 89, further comprising machine executable code that determines said link failure is a failed link between two WAN blades connected by a network.

91. (Previously Presented) The computer program product of Claim 89, further comprising machine executable code that determines said link failure is a failed link between a WAN blade coupled to a primary storage device.

92. (Previously Presented) The computer program product of Claim 91, further comprising: machine executable code that, in response to detecting said failed link, journals writes to the WAN blade rather than the primary storage device, said WAN blade acting as a buffer to compensate for said failed link.

93. (Previously Presented) The computer program product of Claim 92, further comprising machine executable code that, upon said WAN blade having a journal that overflows, causes said WAN blade not to acknowledge write operations by the first storage device.

94. (Previously Presented) The computer program product of Claim 93, further comprising:  
machine executable code that clears said journal on said WAN blade in response to said journal overflowing by sending messages to the primary storage device to invalidate previously written and acknowledged data.

95. (Currently amended) ~~[[The]]~~ A computer program product of Claim 80, for performing data recovery in a computer system, comprising:

machine executable code that sends data from a first storage device to at least one other secondary storage device, said data being sent in a plurality of data packets, each of said plurality of packets being associated with a sequence number having a first predetermined value;

machine executable code that, upon determining that the data has been successfully stored on all of said at least one other storage device, deletes journal entries in a sender corresponding to said data;

machine executable code that, upon determining a failure in connection with synchronizing data between a first storage device and at least one other secondary storage device, deletes journal entries in each of said at least one other secondary storage device, and resends unsynchronized journal entries from the sender by sending a plurality of data packets all having a same sequence number lower than sequence numbers associated with other unsent packets and then sending any remaining data packets having a next higher sequence number, wherein data packets having the same sequence number are sent in an order that is independent of an order in which the data packets were created and wherein data packets having the same sequence number represent different data that is independently generated from a plurality of different sources; and

~~further comprising~~ machine executable code that, in response to the sequence number in the sender becoming equal to a second predetermined value different from the first predetermined value, acknowledges receipt of the blocks of data corresponding to the packets of data that are assigned the first predetermined value as the sequence number and sending the packets of data that are assigned the first predetermined value as the sequence number to said at least one other secondary storage device.

96. (Previously Presented) The computer program product of Claim 95, wherein said machine executable code that acknowledges includes machine executable code that sends an acknowledgement to a host in the computer system sending data to the first storage device prior to said data actually being transferred to the at least one secondary storage device.